Design of Nested Protograph-Based LDPC Codes with Low Error-Floors

Matthew L. Grimes and David G. M. Mitchell
Klipsch School of Electrical and Computer Engineering,
New Mexico State University, Las Cruces, NM 88011
Email: {mattg92,dgmm}@nmsu.edu

***Abstract*—In this paper, we propose a method to design nested protograph-based low-density parity-check (LDPC) codes for network communications with low error-floors via a graph modification procedure. Lowering the error-floor of nested LDPC codes is essential for certain applications that require very low decoded error rates in moderate to high signal-to-noise-ratios (SNRs) like optical communication and data storage. The multi-level method we propose improves upon existing state-of-the-art algebraic designs by the identification and removal of harmful graph structures from nested LDPC codes, effectively improving the performance of each of the nested codes in their error-floor region. Simulation results are provided that confirm the expected performance improvement.**

***Index Terms*—Low-density parity-check (LDPC) codes; nested codes; error-floor; absorbing set, network communication.**

I. Introduction

Linear nested codes [1], where subcodes are embedded in a larger code, were used in the 1970s for reliable storage of information in memory with defective cells [2] and have subsequently been widely applied in various multi-terminal source and channel coding problems (see e.g., [1], and references therein). In particular, we are interested in nested schemes for joint channel and network coding, where codewords of different subcodes C_i, $i = 1, 2, ..., N$, are algebraically superimposed via a bitwise XOR [3]. Such codes were proposed, for example, in [4] for the generalized broadcast relay problem and two-way relaying in [5]. A particular challenge of this problem is that each subcode, and combinations of subcodes, should be itself a good channel code.

Low-density parity-check (LDPC) [6] codes have been shown to exhibit exceptional error-correcting performance with low-complexity iterative belief propagation (BP) decoding. These capabilities have led to the application of LDPC codes in various areas, such as multi-user communications. It is well-known, however, that small graphical sub-structures existing in the Tanner graph of LDPC codes can cause iterative decoding algorithms to get trapped in certain error patterns. These weaknesses contribute significantly to the performance of the code in the so-called *error-floor*, or high SNR, region of the bit error rate (BER) performance curve. For the binary-input additive white Gaussian noise (AWGN) channel, these error-prone sub-structures are named *trapping sets* [7]. In that paper, trapping sets were shown to have a strong influence on both the position and slope of the error floor. Previously, such weaknesses of LDPC codes on the binary erasure channel (BEC) had been defined as *stopping sets* (SS) [8], which can be viewed as trapping sets on the BEC under BP decoding. In [9], the authors proposed an important subclass of trapping sets called *absorbing sets* (AS) that were shown to account for nearly all decoding errors in the error floor of certain LDPC codes. An (a, b) AS is defined as a decoder-independent combinatorial object that can be studied analytically.

The approach we present in this paper involves a graph-based method of targeted graph shortening for removal of harmful objects in particular nested LDPC codes to improve the BER performance in the high SNR region, thereby lowering the error-floor of the global code and all associated subcodes. The technique involves targeting problematic graph structures, such as stopping sets (SS) and/or absorbing sets (AS), and breaking the connectivity of graphical objects by careful modification of the graph. Note that particular care needs to be taken to ensure improvement in all nested subcodes. Our multi-level method is a general way to systematically remove harmful objects via graph modification, or targeted shortening of the LDPC code. We demonstrate the approach as applied to a state-of-the-art algebraic nested code structure [3] and provide simulation results showing lower error-floors of all subcodes in the nested code structure with only a very slight loss in code rate.

II. Protograph-based Nested Codes

In this section, we provide background on Tanner graphs, absorbing sets, protograph-based code construction, and the general concept of nested codes.

A. Tanner graphs and absorbing sets

We associate a bipartite *Tanner graph* with the $m \times n$ parity-check matrix $\mathbf{H}$ of a linear code in the usual way [10]. The set of n variable nodes associated with the columns of $\mathbf{H}$ is denoted by V, and the set of m check nodes associated with the rows of $\mathbf{H}$ is denoted as F. The *girth* of a Tanner graph associated with a parity-check matrix $\mathbf{H}$ is the length of the shortest simple cycle in the graph, i.e., the shortest closed walk with no repetitions of nodes or edges.

For any subset D of the set of variable nodes V, let $E(D)$ (resp. $O(D)$) be the set of neighboring check nodes with even (resp. odd) degree with respect to D. An (a, b) AS is a subset $D \subseteq V$ of size $a \geq 0$, with $O(D)$ of size $b \geq 0$ and with the property that each variable node in D has strictly fewer

neighbors in $O(D)$ than in $F\backslash O(D)$. We say that an (a, b) AS D is an (a, b) *fully AS* (FAS) if, in addition, all nodes in $V\backslash D$ have strictly more neighbors in $F\backslash O(D)$ than in $O(D)$.

B. Protographs

A protograph [11] with *design rate* $R = 1 - n_c/n_v$ is a small bipartite graph that connects a set of n_v variable nodes to a set of n_c check nodes by a set of edges. The protograph can be represented by a parity-check or *base* biadjacency matrix $\mathbf{B}$, where $B_{x,y}$ is taken to be the number of edges connecting variable node v_y to check node c_x. The parity-check matrix $\mathbf{H}$ of a protograph-based LDPC code can be created by replacing each non-zero entry in $\mathbf{B}$ by a sum of $B_{x,y}$ non-overlapping permutation matrices of size $M \times M$ and each zero entry by the $M \times M$ all-zero matrix. It is an important feature of this construction that each derived code inherits the degree distribution and graph neighborhood structure of the protograph. The *ensemble* of protograph-based LDPC codes with block length $n = Mn_v$ is defined by the set of matrices $\mathbf{H}$ that can be derived from a given protograph using all possible combinations of $M \times M$ permutation matrices.

C. Nested Codes

Nested codes are well-suited for various applications in the area of network communications. Nested codes can be simply defined [1] as a pair of linear codes (C_1, C_2) that satisfy the following relation

$$C_2 \subset C_1, \tag{1}$$

which implies that each codeword of C_2 is also a codeword of C_1. This can be extended to multiple subcodes for the same base, or global, code. Nested codes can be defined via a partitioning of the parity check matrix $\mathbf{H} \in \mathcal{F}_2^{m \times n}$, where $\mathbf{H}$ is partitioned into (potentially overlapping) submatrices $\mathbf{H}_i$ of size $m_i \times n$. Then, the resulting subcodes C_i of rate $R_i \geq 1 - m_i/n$ are defined via the null space of their parity check matrices $\mathbf{H}_i$, $i = 1, 2, \ldots, N$, and

$$\mathbf{c}\mathbf{H}^T = \mathbf{0} \Rightarrow \mathbf{c}\mathbf{H}_i^T = \mathbf{0}.$$

Note that if any submatrix $\mathbf{H}_i$ is completely contained within another, $\mathbf{H}_j$, then $C_j \subset C_i$ and $R_j \leq R_i$. Selecting which m_i rows to choose to form code C_i can be a difficult optimization problem; however, by imposing a protograph structure, the problem reduces to choosing subgraphs of the protograph with good ensemble properties.

Protograph-based nested codes are constructed using the procedure described Section II-B. A nested base matrix $\mathbf{B}$ of size $b \times c$ is constructed, and partitioned into submatrices $\mathbf{B}_1, \ldots, \mathbf{B}_N$ of size $b_i \times c$ according to the target design rates of the subcodes $R_i \geq 1 - b_i/c$, $i = 1, 2, \ldots, N$. After graph lifting the nested base matrix, the lifted $\mathbf{B}_i$ corresponds to the parity-check matrix $\mathbf{H}_i$ of size $b_i M \times cM$ corresponding to subcode C_i. The degree distribution of $\mathbf{H}$ and permutations used in the graph lifting can be optimized to construct good lifted codes [3].

III. Multi-level Graph Optimization

In this section, we first review the design example from [3] used to demonstrate our proposed procedure, then apply the graph modification technique.

A. Design Example

For the purpose of demonstrating the procedure and providing numerical results, we study the nested protograph described in [3]. This protograph has a 12×16 base parity-check matrix

$$\mathbf{B} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

After lifting with $M = 305$, the resulting parity-check matrix has size 3660×4880. In this new lifted matrix, the first $b_1 = 10$ row blocks, or rows 1 through 3050, form parity-check matrix $\mathbf{H}_1$ and represent subcode C_1, the middle 8 row blocks, rows 611 through 3050, form $\mathbf{H}$ and represent the "global" code C, and the final $b_2 = 10$ row blocks, rows 611 through 3660, form $\mathbf{H}_2$ and represent subcode C_2. Note that $C_1 \subset C$ and $C_2 \subset C$. During the lifting process, the permutation matrices were carefully selected to avoid 4-cycles, therefore the graphs corresponding to $\mathbf{H}$, $\mathbf{H}_1$, and $\mathbf{H}_2$ all have girth at least 6. The three codes C, C_1, and C_2, all of length $n = 4880$, have code rates $R = 0.5$, $R_1 = 0.375$, and $R_2 = 0.375$, respectively.

Each of these codes was simulated with binary-phase shift-keying (BPSK) modulation on a binary-input AWGN channel using the sum-product decoding algorithm (SPA). The simulations were allowed a maximum of 100 iterations and at least 100 word errors were collected for each SNR point. The resulting BER performance curves for each of the three nested code sections can be seen in Figure 1 (solid lines). Also shown for comparison are randomly constructed codes without 4-cycle prevention (dashed lines). We observe that the three codes have good waterfall performance (due to the good iterative decoding threshold of the protograph); however, there is a clear change of slope at around 1.5 dB to 2.0 dB, where the error floor region begins. In this region, the probability of error decays slowly with SNR, flattening off at BERs around 10^{-5} to 10^{-6}. The poor performance in this region motivates our efforts to optimize the graph to lower the error-floor. Note that the 4-cycle free codes have similar performance to the

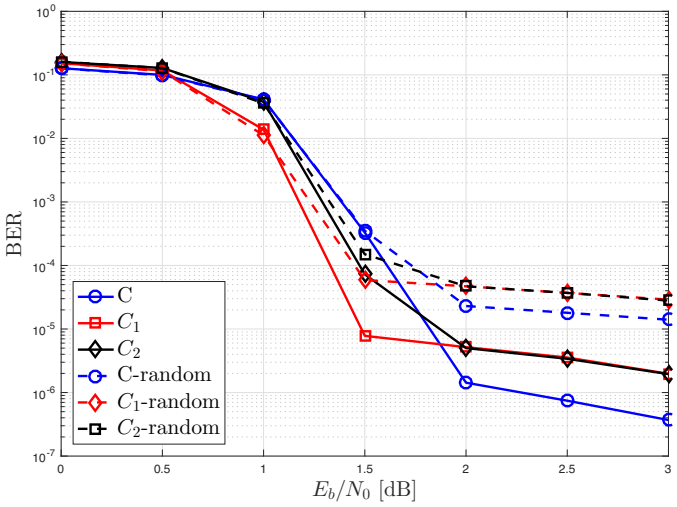


Fig. 1: Performance of protograph-based nested codes C, C_1, and C_2, randomly constructed (dashed lines) and constructed with no 4-cycles (solid lines).

codes presented in [3]. Further efforts to increase the girth of the nested graphs could prove helpful; however, as we show in the next subsection, dominant errors from typical lifted graphs of this construction correspond to a 12-cycle. It is a very challenging problem to lift this protograph such that all nested codes have girth 14.

B. Identification and Removal of Harmful Structures

The first step in our procedure is to determine the dominant errors for each of the subcodes. For certain code constructions, this may be known in advance, e.g., structured array codes with column weight 3 have $(4, 2)$ absorbing sets as the dominant errors when decoded with SPA [9], for other codes this can be determined empirically by computer simulation or by exhaustive search of the graph [12].

For the example nested code structure discussed in Section III-A we performed an empirical investigation via computer simulation to determine which parts of the nested code graphs were causing errors and to identify the nature of these objects. The error locations were recorded by the simulation software and processed to determine local connectivity. The results obtained for this error locating procedure can be seen in Table I, which details the most frequent, or "most dominant error", recorded for a given SNR. The SNR range was tested in 0.5 dB increments from 0 dB up to 3 dB, which captures the standard operational range of these nested codes.

We note that for highly irregular constructions, such as this nested code, we have found that the multiplicity of dominant errors can be small. This is not the case for structured regular codes, e.g., array codes [9], for which our approach may not be as beneficial. Table I shows the specific results for the global code C described in Section III-A. In the case of low SNR, there were no dominant errors of length 10 or less collected; these longer errors consist of a concatenation of multiple trapping sets/AS and are not particularly relevant for our study since they do not occur at moderate to high SNRs where the probability of bit error is lower. We can see that the most frequently observed error location sequence, bit locations $E = \{322, 522, 582, 1969, 2055, 2061\}$, was found in all the simulations at higher SNR points (1.0 dB to 3.0 dB). The third column in the table records the proportion of occurrences of the dominant error. As the SNR increases, we observe for this code that the most recorded error was dominant and there was no clear second most common error. We note that, for SNRs at least 1.5 dB, error sequence E accounts for at least 23% of all errors collected and as high as 93%, which motivated our investigation of methods to eliminate this harmful subgraph via graph modification. The proposed approach of graph modification via shortening works particularly well for structures with clear dominant errors such as this.

The results for nested subcodes C_1 and C_2 were similar to global code C, so their error location tables have been omitted due to space constraints. For these subcodes, we observe that the same subgraph induced by E is present in their graphs and is also the dominant error in both of those simulations. Consequently, elimination of this structure can potentially improve the performance of nested codes C_1 and C_2 (in addition to C), and is not likely to degrade their performance.

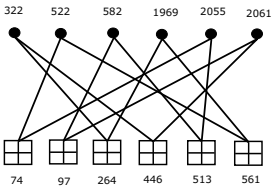
SNR	Most Dominant Error Recorded With Length ≤ 10	No. Occurrences / No. Errors Collected
0.0 dB	No Length 10 Errors Collected	N/A
0.5 dB	No Length 10 Errors Collected	N/A
1.0 dB	{322,522,582,1969,2055,2061}	1/100 (1%)
1.5 dB	{322,522,582,1969,2055,2061}	32/137 (23%)
2.0 dB	{322,522,582,1969,2055,2061}	132/141 (93%)
2.5 dB	{322,522,582,1969,2055,2061}	111/123 (90%)
3.0 dB	{322,522,582,1969,2055,2061}	112/124 (90%)

TABLE I: Error Information for Global Code C

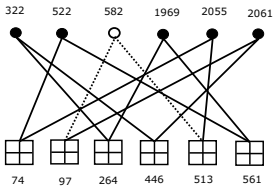
The induced graph in the subgraph of $\mathbf{H}$ according to code C and corresponding to E is shown in Figure 2(a), which corresponds to a 12-cycle, or equivalently a $(6, 0)$ absorbing set/codeword with Hamming weight 6. The variable nodes involved in error location sequence E have degree 2 (i.e., they are localized to the low column weight/weaker region of the graph). We note that this object prevents the SPA from correcting errors at these locations and is also harmful for other channels, such as the BEC, where it can be considered as a stopping set. Unlike regular LDPC codes, this highly irregular nested code does not have uniformly distributed codewords of low weight, in fact this occurrence was the only one found in the construction. This further motivates our approach to seek and eliminate the (few) dominant weaknesses in the nested graph.

C. Graph Modification (Shortening)

Following the identification of errors, we now attempt to improve the performance of the nested construction by sequentially modifying the graph to remove the harmful objects.

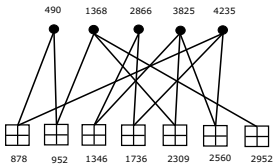


(a) Induced Tanner graph for C

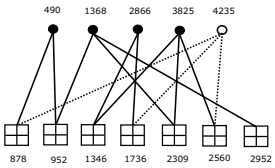


(b) Induced Tanner graph for C' with variable node 582 removed

Fig. 2: Induced Tanner graphs of E for the global code C before and after modification



(a) Induced Tanner graph for E_2



(b) Induced Tanner graph for E_2 with variable node 4235 removed

Fig. 3: Induced Tanner graphs of E_2 before and after modification

We focus on the dominant errors, particularly those that occur throughout the nested code structure, the presence of which result in a relatively high error floor in all nested codes at high SNRs. We revisit the subgraphs present in the graphs of C, C_1, and C_2 induced by the 6 variable nodes in E and shown in Figure 2(a). Note that the check node indices change for subcodes C_1 and C_2, but the graph induced by E has the same structure.

We propose a graph shortening technique to 'break' the object: one variable node is removed from all nested codes, including edges to adjacent check nodes. This has two effects:

1) the code rate is slightly decreased, assuming the rank of $\mathbf{H}$ is unchanged;
2) the graph of all nested codes is modified such that the object is changed.

Figure 2(b) depicts one way to break the cycle by removing one of the nodes in the induced subgraph. In this example, we chose to shorten the code by removing variable node 582 and attached edges from the graph, which is now shown as a white circle with dotted lines indicating prior connection to its check nodes. The decision to remove node 582 was made arbitrarily but due to the symmetry of this structure, similar results are obtained by removing any one of the other variable nodes in E.

Note that, by modifying the nested code/parity-check matrix structure in this way, care needs to be taken to prevent creating new harmful structures in any of the subcodes. For example, we wish to avoid creating degree one variable and/or check nodes, as well as new (smaller) absorbing sets. We note that although the induced graph of E has the same structure in all subcodes, the check nodes involved in each varies, thus care must be taken to ensure good performance in all resulting subgraphs. The removal of variable node 582 does not create any new harmful structures and only slightly lowers the code rates. The original codes C, C_1, and C_2 have rates $R = 0.5$, $R_1 = 0.375$, and $R_2 = 0.375$, respectively, while the modified codes C', C'_1, and C'_2 have rates $R' = 0.4999$, $R'_1 = 0.3749$, and $R'_2 = 0.3749$, respectively.

The graph modification technique can be extended to an arbitrary number of levels. Continuing this example, we performed targeted graph shortening on the already modified codes C', C'_1, and C'_2 at the second level to construct codes C'', C''_1, and C''_2, respectively. A highly dominant error structure, such as that found and removed in the first level of modification was not present at the second level; instead we observe multiple less dominant sequences with the proportion of occurrences of the "dominant" error in the 2% to 9% ranges instead of 23% to 99% like we saw at the first level. Consequently, we do not expect such a dramatic improvement as with the first level.

The global code C' and subcode C'_1 simulations contained the error location sequence $E_2 = \{490, 1368, 2866, 3825, 4235\}$ which is dominant and now targeted for removal. This error location sequence, or a subset of E_2, occurred in a maximum of 4% of the recorded errors in C' and in a maximum of 9% of recorded errors in C'_1 over the error-floor region. Simulations of subcode C'_2 did not show E_2, which leads us to expect little or no improvement in its level two BER performance. Figure 3(a) depicts the induced Tanner graph for E_2. We note that this set of variable nodes corresponds to a $(5, 1)$ AS that contains several cycles of different lengths. Figure 3(b) depicts the induced Tanner graph for E_2 after the removal of variable node 4235 from the parity-check matrices of C', C'_1, and C'_2. Note that removing this node breaks all of the cycles in the subgraph.

As a consequence of removing variable node 4235, the rate of the second level modified codes is further decreased. The first level consisted of codes C', C'_1, and C'_2 with rates $R' = 0.4999$, $R'_1 = 0.3749$, and $R'_2 = 0.3749$, respectively; while the second level consisted of codes C'', C''_1, and C''_2 with rates $R'' = 0.4998$, $R''_1 = 0.3747$, and $R''_2 = 0.3747$, respectively. As will be confirmed empirically in in Section IV, the limited presence of E_2 in the nested codes is not expected to result in as large of an improvement in BER performance as the removal of E had in the first level of graph modification.

The approach can be extended as many times as necessary. Note that if the multiplicity of a particular object is large, graph shortening should be repeatedly applied to eliminate all objects in that class (if possible). This could result in large rate loss. For irregular constructions, such as this design example, this is typically not the case. Finally, we note that similar results to those presented here were obtained from different nested codes from the protograph-based ensemble.

IV. Numerical Results

Figure 4 depicts the results of computer simulations performed on our modified nested code structures C', C'_1, and C'_2 (dashed lines) compared to the performance of the

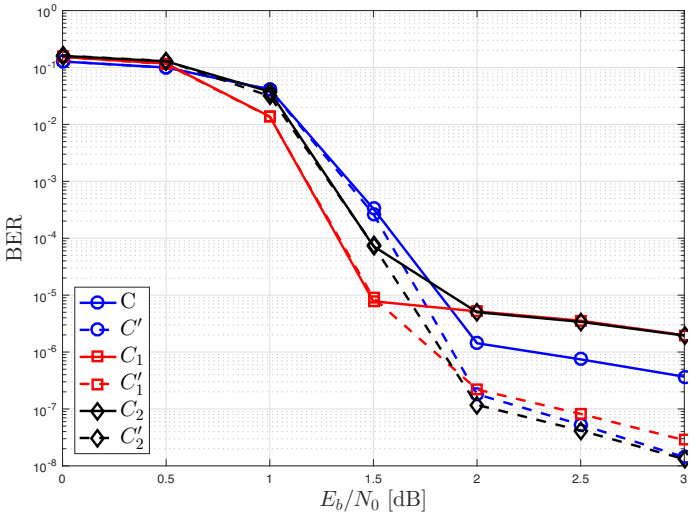


Fig. 4: Original nested codes C, C_1, and C_2 (solid lines) vs. first level modified codes C', C_1', and C_2' (dashed lines)

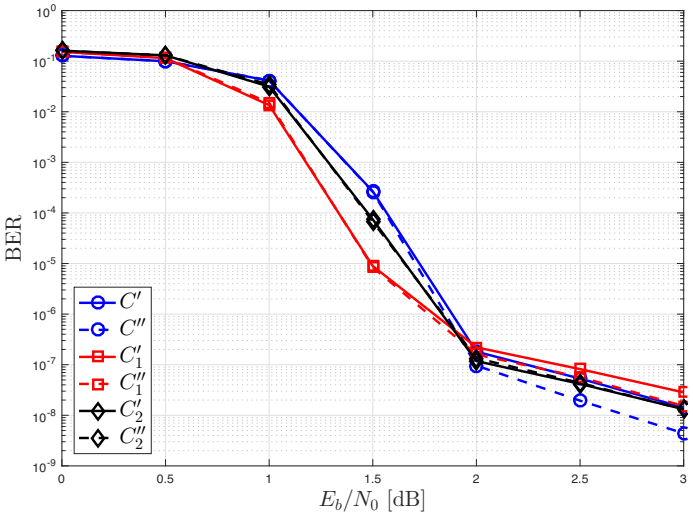


Fig. 5: First level modified codes C', C_1', and C_2' (solid lines) vs. second level modified codes C'', C_1'', and C_2'' (dashed lines).

original codes C, C_1, and C_2 (solid lines), respectively. We observe that the performance of all of the codes after graph modification experience a significant improvement in BER in the higher SNR ranges, 2 dB to 3 dB, which constitute the error-floor regions discussed previously. We also note that the frame error rate (FER), not shown in the figure, also displayed similar performance improvement. This improvement was achieved for only a slight loss in rate, as discussed in Section III-C.

Figure 5 depicts the results of the second level of our graph modification technique. It should be noted that as the graph modification process is applied to already modified codes, the improvement in BER and FER will typically get smaller as the harmful structures in the codes graphs become less prevalent and are more difficult to remove. We can see in Figure 5 that applying the graph modification procedure to the already modified codes C' and C_1' has a noticeable (but less significant than the first level) impact on the BER in the error-floor regions (2 dB to 3 dB) while modifying C_2' did not have much of an impact, with almost identical results to the first level of modification. This was expected since the object removed from C_2' did not appear as a dominant error object in its respective recorded error locations. It should be noted that even though C_2'' did not show a noticeable improvement in the error-floor region, both C' and C_1' were improved - thus further global optimization could be considered.

V. Conclusion

In this paper, a method to lower the error-floor of nested protograph-based LDPC codes was proposed. The method involved the use of a graph shortening procedure that required identification and targeted removal of harmful graph structures, such as absorbing sets. We showed by example that this procedure significantly improved the BER performance in the error-floor region for both the global code and subcodes in the nested code structure. In addition, we demonstrated that this method could be applied repeatedly to further improve the error-floor performance of the nested code structure.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1710920.

References

[1] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, Jun 2002.

[2] A. V. Kuznetsov and B. S. Tsybakov, "Coding in a memory with defective cells," *Problems of Information Transmission*, vol. 10, no. 2, pp. 52–60, Apr. 1974.

[3] C. A. Kelley and J. Kliewer, "Algebraic constructions of graph-based nested codes from protographs," *Proc. IEEE International Symposium on Information Theory, Austin, TX*, Jul. 2010.

[4] L. Xiao, T. E. Fuja, J. Kliewer, and D. J. Costello, "A network coding approach to cooperative diversity," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3714–3722, Oct 2007.

[5] C. Hausl and J. Hagenauer, "Iterative network and channel decoding for the two-way relay channel," in *Proc. IEEE International Conference on Communications*, vol. 4, June 2006, pp. 1568–1573.

[6] R. Gallager, "Low-density parity-check codes," *MIT press*, 1963.

[7] T. Richardson, "Error floors of LDPC codes," *Proc. 41st Annual Allerton Conf. Commun., Control, and Computing, Monticello, IL*, pp. 1426–1435, Oct. 2003.

[8] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-length analysis of low-density parity-check codes on the binary erasure channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, Jun 2002.

[9] L. Dolecek, Z. Zhang, V. Anantharam, M. J. Wainwright, and B. Nikolić, "Analysis of absorbing sets and fully absorbing sets of array-based LDPC codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 1, pp. 181–201, Jan. 2010.

[10] R. Tanner, "A recursive aproach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep 1981.

[11] J. Thorpe, "Low-density parity-check (LDPC) codes constructed from protographs," *Interplanetary Network Progress Report*, vol. 154, pp. 1–7, Aug. 2003.

[12] Y. Hashemi and A. H. Banihashemi, "Characterization and efficient exhaustive search algorithm for elementary trapping sets of irregular LDPC codes," in *Proc. IEEE International Symposium on Information Theory (ISIT), Aachen, Germany*, June 2017, pp. 1192–1196.